

# Workshop

---

## Fakta

Kurs:	Systemutvecklare C/C++ Extended 2024
Klass:	SUVx24
Teknikområde:	Programspråket C, in- och utmatning, funktioner

---

## Learning Target

### Programmering i C och C++

Grundläggande begrepp inom programmering.

Skapa källkod med C syntax. Tema funktioner.

### Grundläggande in- och utmatning

In- och utmatning av resultat från ett program med hjälp av standardfunktioner i C.

### Användning av utvecklingsmiljöer

Använda *Visual Studio Code* (förkortat *VSC*) för att skapa källkod, kompilera och exekvera program skapat med programspråket C.

## Innehåll – Del 2

Träna på att använda funktioner och göra refaktorering.

### Uppgift 5

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-5.c**

Skapa ett program som

- Använder en funktion `cubeVolume()` för att beräkna volymen av en kub.
- I huvudprogrammet frågar efter en sidas längd av kuben (i cm?)
- Använder det inmatade värdet som argument till funktionen.
  - Lägg till nödvändig deklaration av parameter till funktionen.
- Skriver ut beräkningen genom ett funktionsanrop med sidans längd.

# Workshop

---

## Uppgift 6

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-6.c**

Skapa ett program som

- Räknar ut medelvärdet av tre (hel)tal.
- I huvudprogrammet frågar användaren efter tre tal.
- Anropar en funktion `average3()` som tar tre inparametrar.
  - Skicka in de tre värdena i funktionen med ett funktionsanrop.
  - Funktionen räknar ut medelvärdet och returnerar resultatet.
- Skriver ut beräkningen på vilka tal som medelvärdet räknas ut på, samt resultatet från funktionsanropet.

*Extra:* Refaktorera programmet att kunna ta in mer än tre värden och räkna ut medelvärde på dessa.

## Uppgift 7 – Bonus

Om du hinner, prova att lösa denna uppgift genom att återanvända en tidigare uppgift.

**Att göra:**

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-7.c**

Kopiera koden från workshop7-6 alternativt workshop7-7. Refaktorera koden att använda funktioner.

- En funktion som presenterar menyn och returnerar valet av operation.
- En funktion som tar det returnerade menyvalet och använder det att välja rätt operation tillsammans med inmatning av tal.
- En funktion per respektive matematikoperation. Dessa funktioner anropas i föregående funktion när rätt operation identifierats.
- En funktion som presenterar resultatet av beräkningen baserat på vilken operation som valdes.
- En funktion som utför kontroll ifall användaren vill fortsätta att köra programmet.
- Ett huvudprogram där funktionerna anropas ifrån.

Testkör programmet och prova alla alternativen att de fungerar som tänkt.

# Workshop

---

- Prova även med felaktiga inmatningar och inför felkontroll där det behövs.
- Potentiella problem:
  - Vad händer om division med noll utförs?
  - Vad händer när roten ur ett negativt tal utförs? (workshop7-7.c)

# Workshop

---

## Uppgift 7 - Extra

Gör denna uppgift i mån av tid.

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-7.c**

Utöka programmet från **Uppgift 6**.

- Undersök `<math.h>` på webbsidan [C mathematical functions](#). `Math.h` finns i **C Standard Library** och i detta "matte"-bibliotek finns en massa funktioner man kan använda utan att själv behöva skapa dem.
- Försök att integrera funktionerna 'roten ur' och 'upphöjt till' i din kalkylator. När funktionerna används i programmet behövs bara ett inmatat värde.

- *Exempelvis*

```
result = sqrt(inputNumber);
```

- Vad händer om ett negativt tal skrivs in för 'roten ur'? Går det att förebygga?

*Extra:* Lägg till fler funktioner, t ex absolutbelopp, maximum och minimum, etc.